Social Navigator: A Designer's tool for Social Navigation

Martin Svensson
Stockholm University/KTH
Department of Computer and Systems Sciences
Box 1263, SE-164 29 Kista
+46 8 6331500
martins@sics.se

ABSTRACT
In this paper we present a toolkit for building systems that support social navigation. The Social Navigator lets a systems designer incorporate various types of communication between users into a system in an easy way. We show an example were the toolkit was used to facilitate social navigation in an on-line shop for buying groceries.

Keywords
Social navigation, design, toolkit

INTRODUCTION
As the community of computer users is rapidly growing so is the information that we can access and potentially have to process. This means that new systems have to be good at filtering and presenting information to their users. There are several ways to tackle this problem of "information overload". For instance, we could try to make systems more personalized for specific users, or use sophisticated information retrieval techniques to filter information.

We have approached the problem of personalizing the system in a slightly different way. Why not let other people do the personalization and let the system more or less act as an intermediary for connecting the right people and to let them communicate with each other in several ways. We call this social navigation [1]. In indirect social navigation there is no direct communication between users. Following the trails of others or recommending information based on what other users think are both examples of indirect social navigation. On the other hand when a person asks someone for directions this is a typical example of direct social navigation. The rationale for using social navigation as a way to support navigation for a user is based on the fact that this is often the way we navigate in the real world.

The Social Navigator (SN) is a toolkit that designers can use when they want to introduce social navigation into their system. Its focus lies in supporting both direct- and indirect social navigation.

THE SOCIAL NAVIGATOR
Since the SN is a designer's tool some special requirements arise. The idea was that the SN should, in a sense, support contact making between people at various levels. The toolkit should allow people to engage in direct social navigation (e.g. chat) and indirect social navigation (e.g. seeing where other people are). Specifically the toolkit had to:

- Keep track of users and their locations. We wanted it to always be able to give information on who were presently using the system and also where they had been. From this we could then derive all kinds of social navigation support.
- Be general, allowing it to be used for different types of systems, ranging from incorporation into a regular HTML page to be incorporated as a component in a stand-alone application.

Implementation
We decided to work with a client-server solution. The reason for this was two-folded. Firstly, clients only need to keep track of a server, and when a client wants to get in contact with another client it does so through the server. Secondly, it is easier to administrate users (which users are allowed to use it, who is on-line, etc.) from a central location.

One of the requirements was generality, so we had to be careful when designing the server. For instance, we could have built our own server and communication protocol for it, but then we would sacrifice generality, i.e. it would be difficult to access it from a web page without having to build, for instance, a Java applet that can communicate on that specific protocol. Also, the acceptance of a system is likely to be greater if it is based on commonly accepted techniques. In effect we choose a solution based on the HTTP protocol with servlets and SQL-databases to store persistent information. This means that our server can be viewed as a web server with the client using *post* and *get* requests to communicate with it. The server can be any web server that supports servlets.

We believe that the proposed solution satisfies our requirements. We use a well known protocol (HTTP) and the communication is purely text based. In reality this means that any system that can communicate with a web server and interpret text, can use the Social Navigator. However, it

is important to note that this solution demands more work on the client side. Also, by using the HTTP protocol it is not possible to achieve true synchronous communication, e.g. chatting is difficult to achieve.

Functionality

When we turn to the specific functionality that the SN offers we can divide it into three areas: functions for administrating users, functions for direct communication between users, and functions for indirect communication (for example, seeing where others are).

As in a normal system that handles multiple users, the SN provides functions for creating and deleting users. To each user it is also possible to attach a profile and later retrieve it. The profile can be of any length and contain whatever the designer choose, as long as it is in text form. A special log keeps track of the movements of a user, i.e. what locations the user has visited in the past. The log is automatically generated and can be requested by a client. One important aspect of these types of systems is the ability for users to set their visibility to the outside world. The SN supports four visibility modes: "visible", "do not disturb", "be right back", and "invisible". Finally, users can login and logout from the SN.

Supporting direct social navigation is achieved in two ways. Firstly a client can send a private message to another client. Each private message to a client is stored at the server until the client fetches them. In addition a client can always send messages to a specific location. Several other clients can then be listening on that particular location and receive the messages sent by the first client (these requests only returns when there is a new message on that location). In order to achieve a private chat two clients could agree to listen and send messages to a location that only those two clients know of.

The way the SN supports indirect social navigation relies heavily on the ability to see who is logged in and where people are. A client can request information about every user known to the system, and who is logged in and their respective location. We are also thinking of distributing individual user movements, to enable users to see how others have moved.

ON-LINE SHOPPING EXAMPLE

We have successfully used the SN to introduce social navigation features in an on-line shop for food [2]. The on-line shop is essentially a system were a person shops by choosing recipes instead of groceries. In addition the system introduces the notion of "recipe clubs", places with a special theme (e.g. Mexican food) that only contains recipes related to that theme.

Seeing Other People

A major feature of the on-line shop is the visibility of other users. A user can see who is on-line and where they are in the system. We used the SN to implement this. In our on-line shop recipe clubs represent locations; they are placed in an overview map divided into squares representing the different clubs. When a user enters the system she is also logged into the SN. Each time a user changes club the SN is notified of this. Then, at regular intervals the system requests all logged in users from the SN and displays them in the overview map at their respective club (location). In this way we let people see other people and where they currently are.

Chatting With Others

In our on-line store we chose to implement chatting at a "community" level rather than at a personal level, i.e. no private chats are allowed. Again we used recipe clubs, though this time to represent chat rooms. When a user moves to a club the system starts listening for messages in that specific club (room). Also the user can send chat messages within the same club (room).

Viewing User Movements

Every time a user changes location the SN stores this information in a user specific log. In the on-line shop this information is retrieved and presented to the user. In this way a user can always find out which clubs she has previously visited.

CONCLUSIONS

In order to make a toolkit of any kind successful it needs to be easy to use and general enough to fit a variety of needs. We have presented the Social Navigator, which satisfies both these requirements by relying on a web-server approach.

In simulation the SN has been tested to handle around 40 simultaneous users, and since we believe that a system of this kind is not appropriate for a much larger community of logged in users, scalability is not a problem.

REFERENCES

1. Munro, A., Höök, K., and Benyon, D. (eds.). *Social Navigation of Information Space*. Springer Verlag, 1999.
2. Svensson, M., Laaksolahti, J., Höök, K., and Waern, A. A Recipe Based On-line Food Store, in *Conference on Intelligent User Interfaces (IUI'2000)*, (New Orleans Louisiana, Jan 2000), ACM Press.